

#3

Attorney Docket No. 1046.1255

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Takahiro MASUDA, et al.

Application No.:

Group Art Unit:

Filed: June 6, 2001

Examiner:

For: NETWORK BROWSER



**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55**

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-376500

Filed: December 11, 2000

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date(s) as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 6/5/01

By: [Signature]

James D. Halsey, Jr.
Registration No. 22,729

700 11th Street, N.W., Ste. 500
Washington, D.C. 20001
(202) 434-1500

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

09/11/67
J1017 U.S. PTO
09/873368



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2000年12月11日

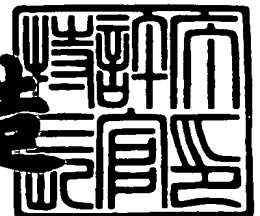
出願番号
Application Number: 特願2000-376500

出願人
Applicant(s): 富士通株式会社

2001年 3月 2日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2001-3013246

【書類名】 特許願

【整理番号】 0000882

【提出日】 平成12年12月11日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/40

【発明の名称】 ネットワークブラウザ

【請求項の数】 10

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 増田 高弘

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 外川 好房

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100089244

【弁理士】

【氏名又は名称】 遠山 勉

【選任した代理人】

【識別番号】 100090516

【弁理士】

【氏名又は名称】 松倉 秀実

【連絡先】 03-3669-6571

【手数料の表示】

【予納台帳番号】 012092

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705606

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 ネットワークブラウザ

【特許請求の範囲】

【請求項 1】 表示画面を管理する表示制御ファイルを解析して別ファイルの記述を抽出する解析手段と、

前記解析手段で得られた前記別ファイルをサーバからダウンロードする手段と、

ダウンロードされた前記別ファイルを不可視画面にロードする手段と、

前記表示画面において表示制御ファイル中の前記別ファイルが指示されたときに、当該別ファイルのダウンロードを行うことなく、前記不可視画面にロードされた前記別ファイルを実行または表示するネットワークブラウザ。

【請求項 2】 前記解析手段は、前記別ファイルがその実行または表示のためにモジュールを必要とするときには、当該モジュールをロードする手段を有する請求項 1 記載のネットワークブラウザ。

【請求項 3】 前記解析手段は、作業用のファイル名をスタックする作業用リストと、

解析が必要なファイル名をスタックする要解析リストと、

解析が不要なファイル名をスタックする解析不要リストとを備え、

前記作業用リストからファイル名を読み出して、当該ファイル名が解析不要リストにスタックされたファイル名と一致していないときには要解析リストにスタックし、要解析リストにスタックされたファイル名を順次読み出して当該ファイル名に基づくサーバへのアクセスを実行させる請求項 1 記載のネットワークブラウザ。

【請求項 4】 前記解析手段は、表示画面への操作を監視し、一定時間以上操作がなされなかったときに前記表示制御ファイルの解析または前記別ファイルのダウンロードを開始する請求項 1 記載のネットワークブラウザ。

【請求項 5】 前記解析手段は、表示画面上の座標指示手段を監視し、当該座標指示手段が一定時間に一定領域内に留まっていたときに前記表示制御ファイルの解析または別ファイルのダウンロードを開始する請求項 1 記載のネットワーク

ブラウザ。

【請求項 6】 表示画面を管理する表示制御ファイルを解析して別ファイルの記述を抽出し、

前記解析で得られた前記別ファイルをサーバからダウンロードし、

ダウンロードされた前記別ファイルを不可視画面にロードし、

前記表示画面において表示制御ファイル中の前記別ファイルが指示されたときに、当該別ファイルのダウンロードを行うことなく、前記不可視画面にロードされた前記別ファイルを実行または表示するネットワークブラウザの表示方法。

【請求項 7】 表示画面を管理する表示制御ファイルを解析して別ファイルの記述を抽出し、

前記解析で得られた前記別ファイルをサーバからダウンロードし、

ダウンロードされた前記別ファイルを不可視画面にロードし、

前記表示画面において表示制御ファイル中の前記別ファイルが指示されたときに、当該別ファイルのダウンロードを行うことなく、前記不可視画面にロードされた前記別ファイルを実行または表示するプログラムを記録したコンピュータ実行可能な記憶媒体。

【請求項 8】 前記解析の際に、前記別ファイルがその実行または表示のためにモジュールを必要とするときには、当該モジュールをロードする請求項 7 記載のコンピュータ実行可能な記憶媒体。

【請求項 9】 前記解析の際に、表示画面への操作を監視し、一定時間以上操作がなされなかったときに前記表示制御ファイルの解析または前記別ファイルのダウンロードを開始する請求項 7 記載のコンピュータ実行可能な記憶媒体。

【請求項 10】 前記解析の際に、表示画面上の座標指示手段を監視し、当該座標指示手段が一定時間に一定領域内に留まっていたときに前記表示制御ファイルの解析または別ファイルのダウンロードを開始する請求項 7 記載のコンピュータ実行可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、インターネット等のネットワーク資源を閲覧するためのブラウザに適用して有効な技術に関する。

【 0 0 0 2 】

【従来の技術】

インターネットWWW(World Wide Web)サイトを閲覧するためのブラウザプログラムとしては、ネットスケープ社(Netscape Corporation)のネットスケープ(Netscape : 商標名)あるいはマイクロソフト社(Microsoft Corporation)のインターネットエクスプローラ(Internet Explorer : 商標名)等が知られている。これらのブラウザでは、HTML(Hyper Text Mark-up Language)という書式で記述されたソースファイルをこれと関連付けられた(リンクが設定された)画像、動画、音声情報等とともに閲覧させる仕組みになっている。

【 0 0 0 3 】

HTMLの特徴は、その記述の仕方でリンクを設定できることであり、これによって階層的なハイパーテキストをネットワーク上で実現している。

【 0 0 0 4 】

したがって、あるページ(HTMLで定義された一つの画面)から別のページに移動する場合には、そのソースページのリンクが定義された部分がマウス等でクリックされることをトリガとして、そのリンク先のファイルの読み込みが開始される。

【 0 0 0 5 】

そのために、リンク先のページが画像サイズの大きい、たとえば1Mbyte以上の大きなファイルの場合、そのファイルを読み込み始めてからブラウザ上に表示し終えるまで数十秒、通信品質によっては数分程度かかる場合もある。

【 0 0 0 6 】

このような点に鑑みて、特開平10-222541号公報に記載されているように、ソースページを指定するとそのソースページからリンクされたHTMLファイルや画像ファイルを事前にローカルな記憶領域に読み込んでしまう技術も提案されている。

【 0 0 0 7 】

しかし、この方式では、ユーザーがソースページを閲覧したときにリンクの実行を指示するか否か不明なファイルまで一括して事前ロードしておくため、バッファ等の記憶領域を無駄に消費してしまう可能性が高かった。

【0008】

さらにリンクされた全てのファイルを先読みするために、当該ページの閲覧が既に完了しているにもかかわらず、当該ページのリンク先ファイルへの先読みが完了していないという現象が起きる可能性があり、先読み処理が実行中のために別のページへの移動（新規なページの読み込み）が遅延する可能性もあった。

【0009】

そのため、この種のファイルの先読み機能は、電話料金の安価な深夜時間帯にあらかじめ決められたソースページを自動巡回するプログラム等でしか用いることができなかった。

【0010】

本発明は、このような点に鑑みてなされたものであり、ソースページの記述内容を解析して当該ファイルの特性に応じて必要なモジュールを事前ロードしたり、カーソル位置や移動状態等から閲覧者の意志を推測して特定の条件の下にファイルの先読みを実行することにより、閲覧者の意志に沿った先読みが可能なブラウザを提供するものである。

【0011】

【課題を解決するための手段】

本発明は、インターネットブラウザ等を用いてHTMLファイル等の表示制御ファイルで規定されたリソースファイルを閲覧する際に、このリソースファイルを解析して、リソースファイルに記述された別ファイルをユーザーから指示がある前にサーバからダウンロードし、当該別ファイルを不可視領域にロードしておくものである。

【0012】

より具体的には、表示画面を管理する表示制御ファイルを解析して別ファイルの記述を抽出する解析手段を設け、この解析で得られた別ファイルをサーバからダウンロードし、このダウンロードファイルを不可視画面にロードしておく、そ



して、前記表示画面で表示制御ファイル中に記述された別ファイルが指示されたときには、あらためて別ファイルのダウンロードは行わずに前記の不可視画面にロードされた別ファイルを実行または表示するようにした。

【0013】

このとき、当該ファイルがモジュールを必要とするときには、当該モジュールをロードするようにしてもよい。

【0014】

また、このような解析処理は、表示画面への操作を監視して、一定時間以上マウスやキーボードの操作がなされなかったときにはじめて実行するようにしてもよい。

【0015】

さらに、マウスカーソルが一定時間に一定領域内に留まっていたときに前記表示制御ファイルの解析または別ファイルのダウンロードを開始するようにしてもよい。ダウンロードするファイルは、同一の上位ディレクトリの下位に配置された複数のファイルを一括して前記サーバからダウンロードしてもよい。

【0016】

【発明の実施の形態】

以下、図面に基づいて、本発明の実施の形態を説明する。

【0017】

【実施例1】

図1は、本発明の一実施例であるブラウザシステムの機能ブロック図、図2は記憶領域の不可視画面保持部とモジュールロード部の領域分割の概念を示したものである。

【0018】

同図中、1はサーバであり、インターネット、すなわちTCP/IP方式によるデータ転送システム上のWWW(World Wide Web)サーバを構成している。

【0019】

本システムは、前記サーバ1にネットワークを介して接続された端末装置であり、パーソナルコンピュータ等で構成されている。

【 0 0 2 0 】

ここで、表示部 1 1 には、図 2 (a) に示すようなリンクの設定されたページが表示されており、この中で指定されているリンク (<http://XX/a.jar>) には、URL の指定ファイル 2 1 a が含まれている。

【 0 0 2 1 】

ユーザーが、表示部 1 1 を通じて所定の操作、たとえば特定の HTML で記述されたホームページ (図 2 (a)) を表示させると、操作内容解析部 1 0 は、当該操作内容を解析する。ここでの解析とは、具体的には、ユーザーの操作によって操作部 1 1 に表示されているページの URL (Uniform Resource Locator) を取得することにある。ここで、図 2 の例では、ファイル 2 1 が取得されることになる。

【 0 0 2 2 】

前記で取得された URL は、バッファである URL 一時蓄積部 3 に一旦蓄積されるが、順次読み出された当該 URL で指示されたサーバ 1 にアクセスされる。次に、前記サーバ 1 より当該 URL のファイル 2 (ここでは、表示部 1 1 で現に表示されているページと同じ URL のソースファイルである) をダウンロードする。このファイル 2 は、ユーザーに意識されることなく、パーソナルコンピュータ上のメモリに展開される。そして、このファイル 2 (表示部 1 1 で表示されている URL のソースファイル) がファイル内容解析部 6 により解析される。

【 0 0 2 3 】

そして解析の結果、先読みすべき URL (図 2 の 2 1 a 等) が存在していた場合には、この URL を URL 一時蓄積部 3 に記憶し、当該 URL に基づいてサーバ 1 へのアクセスを行う。ここで、先読みすべき URL とは、前述の表示部 1 1 に表示されているページのソースファイル中に、リンク先として記述されている URL を意味する。

【 0 0 2 4 】

URL 一時蓄積部 3 に記憶された URL に基づいてサーバ 1 がアクセスされると、当該 URL で指定されたファイル (HTML ファイル) がダウンロードされてこの内容がファイル内容解析部 6 によって解析される。ファイル内容解析部 6

は、このファイル（HTMLファイル）を解析して、動画ファイルや音声ファイル等のように再生モジュールが必要なファイルの指定があるか否かを判定する。この種のモジュールが必要な場合には、モジュール検索部4およびモジュールロード部5を通じて、当該モジュールをパーソナルコンピュータの記憶領域に読み込んでおく。図2（d）の例では、ジャバ用のj a rモジュール24 a、音声用のa i f fモジュール24 bがロードされている。

【0025】

次に、ファイル内容解析部6は、前記でダウンロードしたファイルを不可視画面保持部7に展開しておく。ここで、不可視画面保持部7と、モジュールロード部5は、パーソナルコンピュータの記憶領域に、図2（c）および（d）に示すように設定されている。すなわち、不可視画面保持部7は不可視画面領域23として、モジュールロード部5はモジュールロード領域24として確保されている。

【0026】

図2（d）では、ジャバ用のファイル（j a r）、音声用のファイル（a i f f）、文字表示用のファイル（h t m l）が不可視画面領域23（不可視画面保持部7）に展開されている。

【0027】

表示部11に前述のソースURLに基づくページ（図2の（b））が表示されており、不可視画面保持部7へのファイルの展開が完了していない段階で、ユーザーのマウス操作等でファイル内容解析部6が解析しているファイルが指定された場合には、ファイル内容解析部6は当該ファイルを不可視画面保持部7に展開することなく、直ちに画面イメージ保持部8に展開する。

【0028】

一方、表示部11で表示されているのが前述のソースURLのページ（図2の（b））のままであるときには、ファイル内容解析部6は前述の通り、当該ファイルを不可視画面保持部7に展開しておく（図2（d））。

【0029】

そして、当該ファイルの不可視画面保持部7への展開後に、表示部11におい

てユーザーのマウス操作等により当該ファイルが指定されると、操作内容解析部 10 は、サーバ 1 へのアクセスを行わずに、不可視画面保持部 7 に保持されているファイルを画面イメージ保持部 8 に展開する。これによって、あらかじめ先読みされたファイルが表示部 11 に直ちに表示される。

【0030】

次に、操作内容解析部 10 とファイル内容解析部 6 とが行う解析処理について、図 3、図 4 および図 5 に基づいて説明する。

【0031】

操作内容解析部 10 は、まず要解析リスト 403、解析不要リスト 404 および作業用リスト 402 を初期化（空の状態に）する。ここで、要解析リスト 403 は解析が必要な URL を順次蓄積していくリストである。解析不要リスト 404 は既にダウンロードが完了し、再度の解析が必要ない URL（ファイル）を登録しておくリストである。また、作業リスト 402 は作業用のリストである。これらのリストはいずれ同じ形式であり、図 5 に示すようなテーブル形式で URL（ファイル）名が登録可能となっている。

【0032】

解析の前段階として、まず、基準となる URL（表示部 11 に表示されているページの URL）を空状態の作業用リストに書き込む。

【0033】

以降の処理は、制御処理（図 3（a））とダウンロード処理（図 3（b））に分かれて行われる。

【0034】

制御処理では、まず、作業用リスト 402 が空であるか否かが判定される（ステップ 301）。ここで、前述のように初期化された直後の場合には作業用リスト 402 は空状態であるため、要解析リスト 403 が空か否かが判定される（302）。以上の結果、作業用リスト 402 も要解析リスト 403 も空であり、かつダウンロード処理も行われていないとき（303）には、処理を完了する。

【0035】

一方、作業用リスト 402 に URL が蓄積されているときには、蓄積されてい

る作業用リスト402の先頭のURLを読み出して(306)、これが蓄積不要リスト404に登録されたものでないかどうかを検査する。ここで、解析不要リスト404に合致せず(307)、解析が必要なものであると判定されたときには、現在解析を行っているかないかを判定し(308)、解析していないものであるときには、当該URL(ファイル)を要解析リスト403に追加して(309)、ステップ301に戻る。

【0036】

このようにして要解析リストにURL(ファイル)が追加されると、ステップ302の判定処理で否定技に分岐し、一定時間の待機(305)となる。

【0037】

ダウンロード処理(図3(b))では、まず要解析リスト403がチェックされて(311)、ここで先ほどのようにURL(ファイル)が登録されているときには、この要解析リスト403の先頭のURL(ファイル2)がロードされて、URL一時蓄積部3に登録されて、当該URL(ファイル2)に基づくサーバ1へのアクセスが行われる(317)。そして当該URL(ファイル2)のダウンロード処理406が実行されると、このURL(ファイル2)はもはや解析済みとなるため、解析不要リスト404の末尾に追加される(318)。

【0038】

次に、ファイル内容解析部6は、前記ステップ317でダウンロードしたURL(ファイル2)の文書解析処理407を実行する。具体的には、ファイル内容をチェックして、これが画像ファイル等のように内部の解析を必要とするか否かをチェックするものである。ここで解析が不要である場合には、前記のように当該URL(ファイル2)を解析不要リスト404に追加した後、ステップ311に復帰する(319)。

【0039】

一方、ダウンロードしたURL(ファイル2)を解析してジャバファイル等のように内部にURLの記述が存在する場合には、これらのURL(ファイル)を全て作業用リスト402の末尾に追加する(320)。この結果、次の制御処理(図3(a))でこのファイルが作業対象として要解析リスト403に加えら

れることになる。

【0040】

次に、操作内容解析部10またはファイル内容解析部6が行う文書解析処理（407）の具体例について説明する。

【0041】

当該文書解析処理では、f t p、n n t p等のように、h t t p以外のプロトコルを利用するリソースについては、先読みの必要がないため、当該先読みの対象とはしない。

【0042】

URLは一般に「[プロトコル名] : // [サーバ名] [: ポート] / [パス名]」の形式、具体的には、「http://www.fujitsu.co.jp/xxx/yyy/zzz.html」というような形式で記述されるため、必要となるプロトコルはURLを解析することで知ることが可能である。

【0043】

このようなURLのリソースを解析する場合に、HTMLまたはHTMLに含まれている各種スクリプトのように、そのままテキスト情報として閲覧可能な文書形式の場合には、URLとして記述されている部分を抽出すればよい。たとえば、「」というように記述があった場合、この「http://www.xxx.com/yyy/zzz.html」がURL（ファイル）として抽出される。

【0044】

直接テキスト情報として閲覧できないようなバイナリファイルに関しては、以下のような解析が可能である。

【0045】

JAVA形式での「*.class」または「*.jar」のようなファイル形式の場合、内部にURL記述を含んでいる可能性がある。このようなJAVAクラスファイルの他、マクロスクリプトを含んだ各種文書ファイルやActive X形式のサーバ・クライアント関係実行型のファイルもこれと同様に内部にURL記述を含んでいる可能性が高い。

【0046】

JAVAのクラスファイルは図6に示すような構造を有しているが、この中の定数格納領域に正当なURLのパターンと一致した値があるか否かを解析して、一致したもののロードすべきURLとみなす。図6では「constant#pool」の部分を解析すればよい。

【0047】

URL記述を含んでいる可能性がある他の形式のファイルについても、その多くは文字列定数としてURLを保持している場合がほとんどであるので、該当する箇所の抽出は文字列のパターンマッチングにより検出可能である。

【0048】

「*.jpg」、「*.bmp」、「*.gif」等の画像ファイルのように、内部にURL記述が含まれる可能性の極めて低いファイルについては、解析を行う必要はない。このような画像ファイルの他に、動画ファイル(mov,mpg等)、音声ファイル(wav,mid等)についても同様に解析不要としてもよい。

【0049】

HTML文書中にタグ形式で表記された部分は、文書解析処理407の解析対象となる。HTML文書は、「タグ(コメントを含む)」と「通常のテキスト」の2つの部分で構成されている。スクリプトやオブジェクトで記述されていても、それらは常にタグかテキストかのどちらかになる。テキスト部分の記述は、ユーザーがブラウザプログラムを通して視覚的に認識できる部分であり、タグはそのテキストの部分的な連携属性としてリンク先やファイルを指定する記述部分であるため、本実施形態ではタグだけを解析すればよい。

【0050】

タグはさらに、「タグ名」、「属性名」、「属性値」、「その他の部分(コメント、=, ", / 等)」に分割することができる。具体的には「」というような記述である。この例では、「img」がタグ名であり、「src」と「alt」が属性名、「"pic.jpg"」と「"picture"」が属性値である。

【0051】

これらの中の属性名に関しては、さらに「対応する属性値がURLになりえないもの」、「対応する属性値がURLかもしれないもの」、「対応する属性値が必ずURLになるもの」とに分けることができる。したがって、URLの可能性がある属性値についてのみパターンを解析し、URLと一致した場合に該当するリソース（ファイル）をサーバ1からダウンロードすればよい。

【0052】

図7はジャバスクリプトの記述例である。同図は、あらかじめフォーム（Form）を設けておき、このフォーム内のオブジェクト操作により特定のリンク先への移動を実現させたものである。ジャバスクリプトでリンクを実現する場合、図7の下線部で示すように、リンク先のURLは文字列定数の形で与えられることが多い。この例では、「http://」で開始される文字列をパターンマッチングで抽出することによってリンク先の解析が可能である。

【0053】

【実施例2】

図8は、本発明の別の実施例を示す機能ブロック図である。

【0054】

本実施例は、表示画面上のカーソルの停止時間を検出してURL（ファイル）の先読みを開始する技術である。

【0055】

本実施例において、URL一時蓄積部3に蓄積されたURLに基づく先読み時のサーバ1へのアクセスは、無操作検出タイマ81が一定時間を計測した後に開始される。すなわち、本実施例では無操作検出タイマ81を有しており、該タイマ81は、操作内容解析部10によってリセットされる。操作内容解析部10は、表示部11に対するユーザーのマウス操作を監視しており、マウスによるカーソル93の移動が停止されるとタイマ81の計数をリセットする。タイマ81は、カーソル移動がされない状態が一定時間（たとえば10秒）以上経過すると、トリガ信号を先読みリンク一覧保持部82に出力し、保持部82に保持されたURLをURL一時蓄積部3に通知させる。一時蓄積部3は、この通知にしたがってサーバ1へのアクセスを実行する。

【 0 0 5 6 】

なお、操作内容解析部 1 0、ファイル内容解析部 6 の処理は図 1 ないし図 7 に示した例と同じであるので説明は省略する。

【 0 0 5 7 】

図 9 (a) は、表示部 1 1 上においてカーソル 9 3 が一定時間の移動しない状態を示している。これを無操作検出タイマ 8 1 が検出して、先読みすべき URL (ファイル 9 2) をサーバ 1 からダウンロードして、当該 URL (ファイル 9 2) が不可視画面保持部 7 を構成する不可視画面領域 7 a に展開された状態を示しているのが図 9 (b) である。

【 0 0 5 8 】

【実施例 3】

図 1 0 は、本発明の他の実施例を示す機能ブロック図である。

【 0 0 5 9 】

本実施例は、一定時間内にカーソルの移動がなされたとしてもそれが表示画面上の一定領域内である場合に、URL (ファイル) の先読みを実行する技術である。

【 0 0 6 0 】

本実施例 3 では、カーソル位置認識部 1 0 0 1 を有しており、表示画面 1 1 0 1 上でのカーソル 1 1 0 2 が指示する位置座標が常に認識されている。そして、本実施例では、画面を分割した各領域と当該領域内にあるリンク先とを対応付けた対応テーブル 1 0 0 2 を有している。この対応テーブル 1 0 0 2 には、具体的には、図 1 1 (a) で示した特定領域 1 1 0 3 と、この特定領域 1 1 0 3 内に記述されたリンク先 (URL) との対応関係が示されている。具体的には図 1 1 (c) に示すように、特定領域 1 1 0 3 が、表示部 1 1 の矩形領域の左上の開始点および右下の終了点が x y 座標で規定されており、この領域内に含まれる URL (ファイル) が格納されている。この対応テーブル 1 0 0 2 は、カーソル 1 1 0 2 が一定以上移動する毎に更新される。具体的には移動先のカーソル位置の座標に基づいて画面イメージ保持部 8 から読み出したリソースに基づいて対応テーブル 1 0 0 2 が書き換えられる。

【 0 0 6 1 】

操作内容解析部 1 0 は、カーソル位置認識部 1 0 0 1 と対応テーブル 1 0 0 2 との情報から特定領域内にカーソルがあるか否か（1 0 0 3）、およびタイマ（図 1 0 には図示せず、図 8 のタイマ 8 1 を用いることができる）からの情報により、当該特定領域 1 1 0 3 内でカーソル 1 1 0 2 が一定時間の間とどまっているか否かを検査し（1 0 0 4）、留まっている場合には、対応テーブル 1 0 0 2 を参照してその特定領域 1 1 0 3 で記述されている URL（ファイル）を URL 一時蓄積部 3 に通知する。URL 一時蓄積部 3 は当該 URL（ファイル）に基づいてサーバ 1 へのアクセスを行う。この結果、特定領域 1 1 0 3 内にある URL（ファイル 9 2）がダウンロードされて図 1 1（b）に示すように、不可視画面保持部 7 を構成する不可視画面領域 7 a に展開される。このときのファイル内容解析部 6 による処理は前述の実施例 1 および実施例 2 と同様であるので説明を省略する。

【 0 0 6 2 】

【実施例 4】

本実施例は、画面そのものが複数存在する場合、たとえばフレームで区切られた画面表示の場合のカーソルが位置するフレーム内に存在する URL（ファイル）を先読みする技術である。

【 0 0 6 3 】

本実施例では、図 1 2 に示すように、画面上に分割された各フレームと、このフレーム内で記述された URL（ファイル）との対応関係を規定した対応テーブル 1 2 0 1 を有している。この対応テーブル 1 2 0 1 は、図 1 3（c）に示すように、表示部 1 1 に表示された表示画面 1 3 0 1 の 4 つに分割されたフレーム画面（URL-A、URL-B、URL-C、URL-D）のそれぞれに記述された URL（ファイル）が登録されるようになっている。

【 0 0 6 4 】

本実施例では、カーソル位置認識部 1 0 0 1 でカーソル 1 3 0 2 の位置が認識されると、当該カーソル 1 3 0 2 が位置しているフレーム（図 1 3（a）では URL-B）に含まれる URL（ファイル）が対応テーブル 1 2 0 1 より検出され

る (1 2 0 2) 。

【 0 0 6 5 】

前記で検出された URL (ファイル) は、URL 一時蓄積部 3 に格納されて、サーバ 1 へのアクセスが実行される。この結果、URL-B に記述された URL (ファイル 9 2) がダウンロードされて図 1 3 (b) に示すように、不可視画面保持部 7 を構成する不可視画面領域 7 a に展開される。このときのファイル内容解析部 6 による処理は前述の実施例 1 ないし実施例 3 と同様であるので説明を省略する。

【 0 0 6 6 】

【実施例 5】

本実施例は、同一のディレクトリ下にあるファイルを先読む技術である。ここで、ディレクトリとは、ファイルまたはフォルダで用いられる階層構造を指し、同一の階層にあるファイルを先読みするという意味である。

【 0 0 6 7 】

本実施例 5 では、図 1 4 に示すように、URL 一時蓄積部 3 からサーバ 1 にアクセスしたときに、当該サーバ 1 の URL よりディレクトリとファイルのリストが対応テーブル 1 4 0 1 に取得される。

【 0 0 6 8 】

この対応テーブル 1 4 0 1 を示したものが図 1 5 (c) である。同図に示すように、ディレクトリ格納ファイル名対応テーブル 1 4 0 1 を有している。この対応テーブル 1 4 0 1 には、ディレクトリ (ここでは、http://xx/) とこれに含まれるファイル (b.html, c.html, d.html, e.html) が対応付けられている。

【 0 0 6 9 】

次に、ディレクトリ保持部 1 4 0 2 は、前記対応テーブル 1 4 0 1 から読み出したディレクトリを保持しつつ、サーバ 1 にアクセスしてディレクトリに含まれるファイルを一括してダウンロードする。このようにしてダウンロードされたファイルはファイル内容解析部 6 で解析されて不可視画面保持部 7 または画面イメージ保持部 8 にロードされる。この処理は、前述の実施例 1 ないし 4 と同様であるので説明を省略する。

【 0 0 7 0 】

図 1 5 (a) は表示部 1 1 の表示画面 1 5 0 1 を示しており、同図 (b) は不可視画面領域 7 a にロードされた同一ディレクトリ (<http://xx/>) に含まれるファイル (1 5 0 2 a 等) をロードした状態を示している。

なお、以上の説明では、対応テーブル 1 4 0 1 で対応付けられるファイルは、上位が同一ディレクトリ (<http://xx/>) の直下の階層のファイルだけとした。しかし、図 1 6 (c) に示すように、対応テーブル 1 4 0 1 b は、上位のディレクトリ (<http://xx/>) の下位にある全てのディレクトリに存在するファイル (b.html, YY/c.html, YY/ZZ/b.html, VV/e.html) を対応付けてもよい。この場合には、不可視画面領域 7 a には、図 1 6 (b) に示すように、同じ上位ディレクトリ (<http://xxx/>) の下位のファイル 1 6 0 2 が全て先読みされてロードされた状態となる。

【 0 0 7 1 】

このように、同一ディレクトリの下位に設定されているファイルは、その階層が異なっても、ユーザーの操作によって読み出される可能性は高いため、不可視画面領域 7 a にロードしておくことにより、表示部 1 1 での効率的な表示が可能となる。

【 0 0 7 2 】

なお、図 1 7 は表示部 1 1 での表示画面 1 7 0 1 と、不可視画面領域 7 a でのファイルのロード状態の変化を示したものである。図 1 7 (b) の左図は先読みファイルとして、<http://xx/a.html> と <http://xx/b.html> の 2 ファイルが読み込まれているが、これが表示部 1 1 の表示内容がユーザーの操作によって更新された場合に、新たに 3 つのファイル (<http://xx/a.html>, <http://xx/b.html>, <http://xx/c.html>) を不可視画面領域 7 a に先読みする必要性が生じたとする。このときのファイルの先読みの条件は前述の実施例 1 ないし 5 で説明した場合のいずれかであるとする。この場合、不可視画面領域 7 a には既に 2 つのファイル (<http://xx/a.html>, <http://xx/b.html>) が読み込まれているため、これらの 2 つのファイルは再読み込みを行わない。そして、不可視画面領域 7 a に未だ読み込まれていないファイル (<http://xx/c.html>) のみを読み込む。

【0073】

このように、不可視画面領域7aは、表示部11の表示内容毎に全てリフレッシュするのではなく、表示変更前の画面を基準に読み込まれた先読みファイルと、表示変更後の画面を基準に読み込まれる先読みファイルとの間に、重複するファイルが存在している場合には、重複ファイル以外のファイルを読み込むようにすればさらに表示の高速化を実現できる。

【0074】

以上に説明した各実施例では、ファイルの先読みは表示部11の画面を閲覧しているユーザーにその先読みを意識させることなく、不可視画面保持部7（不可視画面領域7a）に予想される先読み込みファイルをあらかじめロードしておく場合で説明したが、このようなファイルの先読みを画像表示の変化を通じてユーザーに通知するようにしてもよい。

【0075】

図18（a）はその一例であり、ユーザーのマウス操作によりカーソル1801が先読みすべきURL（別ファイル）に近づくと、先読みの実行を通知するためにカーソルそのものをフラッシュ表示（点滅表示）させるようにしている。一方、同図（b）に示すように、カーソルの形状そのものを矢印図形から指の図形（1801a）に変更させるようにしてもよい。このようなカーソル1801の視覚的变化は、操作内容解析部10がカーソル位置認識部1001からの情報に基づいてカーソル位置を認識して、複数のカーソル形状が格納されたキャラクタコードの先頭アドレスを変更することにより容易である。

【0076】

以上、実施例に基づいて本発明を説明したが、本発明は前記実施例に限定されるものではなく、以下に列挙した付記を含む概念である。

【0077】

（付記1） 表示画面を管理する表示制御ファイルを解析して別ファイルの記述を抽出する解析手段と、

前記解析手段で得られた前記別ファイルをサーバからダウンロードする手段と

ダウンロードされた前記別ファイルを不可視画面にロードする手段と、
前記表示画面において表示制御ファイル中の前記別ファイルが指示されたときに、当該別ファイルのダウンロードを行うことなく、前記不可視画面にロードされた前記別ファイルを実行または表示するネットワークブラウザ。

【 0 0 7 8 】

(付記 2) 前記解析手段は、前記別ファイルがその実行または表示のためにモジュールを必要とするときには、当該モジュールをロードする手段を有する付記 1 記載のネットワークブラウザ。

【 0 0 7 9 】

(付記 3) 前記解析手段は、作業用のファイル名をスタックする作業用リストと、

解析が必要なファイル名をスタックする要解析リストと、

解析が不要なファイル名をスタックする解析不要リストとを備え、

前記作業用リストからファイル名を読み出して、当該ファイル名が解析不要リストにスタックされたファイル名と一致していないときには要解析リストにスタックし、要解析リストにスタックされたファイル名を順次読み出して当該ファイル名に基づくサーバへのアクセスを実行させる付記 1 記載のネットワークブラウザ。

【 0 0 8 0 】

(付記 4) 前記解析手段は、要解析リストにスタックされたファイル名に基づいてサーバにアクセスし、当該ファイルをダウンロードした後は、前記ファイル名を解析不要リストにスタックする付記 3 記載のネットワークブラウザ。

【 0 0 8 1 】

(付記 5) 前記解析手段は、表示画面への操作を監視し、一定時間以上操作がなされなかったときに前記表示制御ファイルの解析または前記別ファイルのダウンロードを開始する付記 1 記載のネットワークブラウザ。

【 0 0 8 2 】

(付記 6) 前記解析手段は、表示画面上の座標指示手段を監視し、当該座標指示手段が一定時間に一定領域内に留まっていたときに前記表示制御ファイルの

解析または別ファイルのダウンロードを開始する付記 1 記載のネットワークブラウザ。

【 0 0 8 3 】

(付記 7) 前記一定領域と表示制御ファイルから抽出されたこの領域内に記述されたファイル名とを対応付ける対応テーブルを有しており、前記解析手段はダウンロードすべき前記別ファイルを当該テーブルを参照して決定する付記 6 記載のネットワークブラウザ。

【 0 0 8 4 】

(付記 8) 前記一定領域は、フレームによって分割された各画面であり、前記対応テーブルは各画面と各画面毎に設定された表示制御ファイルに記述された別ファイルとを対応付けたものである付記 7 記載のネットワークブラウザ。

【 0 0 8 5 】

(付記 9) 前記解析手段は、前記別ファイルとして同一の上位ディレクトリの下位に配置された複数のファイルを一括して前記サーバからダウンロードする付記 1 記載のネットワークブラウザ。

【 0 0 8 6 】

(付記 1 0) 前記別ファイルは同一ディレクトリの下位に配置された同一階層のファイルの他、異なる下位の階層に配置されたファイルを含む付記 9 記載のネットワークブラウザ。

【 0 0 8 7 】

(付記 1 1) 前記解析手段は、上位ディレクトリとその下位階層にあるファイルとを対応付けた対応テーブルを有しており、ダウンロードする別ファイルとしてのファイルを当該対応テーブルを参照して決定する付記 9 または 1 0 記載のネットワークブラウザ。

【 0 0 8 8 】

(付記 1 2) 前記不可視画面は前記表示画面を制御する表示制御ファイルが変更される度に更新されるとともに、更新変更前にロードされた別ファイルと、表示変更後にロードされる別ファイルとが同一である場合には更新を行わない付記 1 記載のネットワークブラウザ。

【 0 0 8 9 】

（付記 1 3） 前記解析手段は、前記表示制御ファイルの解析または別ファイルのダウンロードを開始するときには表示画面を変化させて視覚的な通知を行う付記 1 記載のネットワークブラウザ。

【 0 0 9 0 】

（付記 1 4） 前記表示画面の変化は、表示画面上に表示された座標指示手段の視覚的变化である付記 1 3 記載のネットワークブラウザ。

【 0 0 9 1 】

（付記 1 5） 表示画面を管理する表示制御ファイルを解析して別ファイルの記述を抽出し、

前記解析で得られた前記別ファイルをサーバからダウンロードし、

ダウンロードされた前記別ファイルを不可視画面にロードし、

前記表示画面において表示制御ファイル中の前記別ファイルが指示されたときに、当該別ファイルのダウンロードを行うことなく、前記不可視画面にロードされた前記別ファイルを実行または表示するネットワークブラウザの表示方法。

【 0 0 9 2 】

（付記 1 6） 前記解析の際に、前記別ファイルがその実行または表示のためにモジュールを必要とするときには、当該モジュールをロードする付記 1 5 記載のネットワークブラウザの表示方法。

【 0 0 9 3 】

（付記 1 7） 前記解析の際に、表示画面への操作を監視し、一定時間以上操作がなされなかったときに前記表示制御ファイルの解析または前記別ファイルのダウンロードを開始する付記 1 5 記載のネットワークブラウザの表示方法。

【 0 0 9 4 】

（付記 1 8） 前記解析の際に、表示画面上の座標指示手段を監視し、当該座標指示手段が一定時間に一定領域内に留まっていたときに前記表示制御ファイルの解析または別ファイルのダウンロードを開始する付記 1 6 記載のネットワークブラウザの表示方法。

【 0 0 9 5 】

(付記 1 9) 表示画面を管理する表示制御ファイルを解析して別ファイルの記述を抽出し、

前記解析で得られた前記別ファイルをサーバからダウンロードし、

ダウンロードされた前記別ファイルを不可視画面にロードし、

前記表示画面において表示制御ファイル中の前記別ファイルが指示されたときに、当該別ファイルのダウンロードを行うことなく、前記不可視画面にロードされた前記別ファイルを実行または表示するプログラムを記録したコンピュータ実行可能な記憶媒体。

【 0 0 9 6 】

(付記 2 0) 前記解析の際に、前記別ファイルがその実行または表示のためにモジュールを必要とするときには、当該モジュールをロードする付記 1 9 記載のコンピュータ実行可能な記憶媒体。

【 0 0 9 7 】

(付記 2 1) 前記解析の際に、表示画面への操作を監視し、一定時間以上操作がなされなかったときに前記表示制御ファイルの解析または前記別ファイルのダウンロードを開始する付記 1 9 記載のコンピュータ実行可能な記憶媒体。

【 0 0 9 8 】

(付記 2 2) 前記解析の際に、表示画面上の座標指示手段を監視し、当該座標指示手段が一定時間に一定領域内に留まっていたときに前記表示制御ファイルの解析または別ファイルのダウンロードを開始する付記 1 9 記載のコンピュータ実行可能な記憶媒体。

【 0 0 9 9 】

【発明の効果】

本発明によれば、ソースページの記述内容を解析して当該ファイルの特性に応じて必要なモジュールを事前ロードしたり、カーソル位置や移動状態等から閲覧者の意志を推測して特定の条件の下にファイルの先読みを実行することにより、閲覧者の意志に沿った先読みが可能なブラウザが実現できる。

【図面の簡単な説明】

【図 1】 本発明の実施例 1 の機能ブロック図

【図 2】 実施例 1 の表示画面と不可視画面領域の状態を示す説明図

【図 3】 実施例 1 の制御処理とダウンロード処理の手順を示すフロー図

【図 4】 実施例 1 の作業リストと、要解析処理リストと、解析不要リストとの間のファイル名スタックの順序を示すフロー図

【図 5】 実施例 1 の各リストの形式を示す説明図

【図 6】 実施例 1 のジャバのクラスファイルの記述例

【図 7】 実施例 1 のジャバスクリプトの記述例

【図 8】 実施例 2 の機能ブロック図

【図 9】 実施例 2 の表示画面と不可視画面領域の状態を示す説明図

【図 1 0】 実施例 3 の機能ブロック図

【図 1 1】 実施例 3 の表示画面と不可視画面領域と対応テーブルの状態を示す説明図

【図 1 2】 実施例 4 の機能ブロック図

【図 1 3】 実施例 4 の表示画面と不可視画面領域と対応テーブルの状態を示す説明図

【図 1 4】 実施例 5 の機能ブロック図

【図 1 5】 実施例 5 の表示画面と不可視画面領域と対応テーブルの状態を示す説明図

【図 1 6】 実施例 5 の変形例における表示画面と不可視画面領域と対応テーブルの状態を示す説明図

【図 1 7】 実施例において、不可視画面領域の更新の前後の状態を示す説明図

【図 1 8】 実施例において、表示画面またはカーソルの変化を示す説明図

【符号の説明】

- 1 サーバ
- 2 ファイル
- 3 URL一時蓄積部
- 4 モジュール検索部

- 5 モジュールロード部
- 6 ファイル内容解析部
- 7 不可視画面保持部
- 8 画面イメージ保持部
- 1 0 操作内容解析部
- 1 1 表示部
- 2 1, 2 2 表示画面
- 2 1 a 表示画面
- 2 3 不可視画面領域
- 2 4 モジュールロード領域
- 2 4 a, 2 4 b モジュール
- 8 1 無操作検出タイマ
- 8 2 先読みリンケージ一覧保持部
- 9 1 a, 9 1 b 表示画面
- 4 0 2 作業用リスト
- 4 0 3 要解析リスト
- 4 0 4 解析不要リスト
- 1 0 0 1 カーソル位置認識部
- 1 0 0 2 対応テーブル
- 1 1 0 1 表示画面
- 1 1 0 2 カーソル
- 1 1 0 3 特定領域
- 1 2 0 1 対応テーブル
- 1 3 0 1 表示画面（フレーム分割）
- 1 3 0 2 カーソル
- 1 4 0 1 対応テーブル
- 1 4 0 2 ディレクトリ保持部
- 1 5 0 1 表示画面
- 1 6 0 1 表示画面

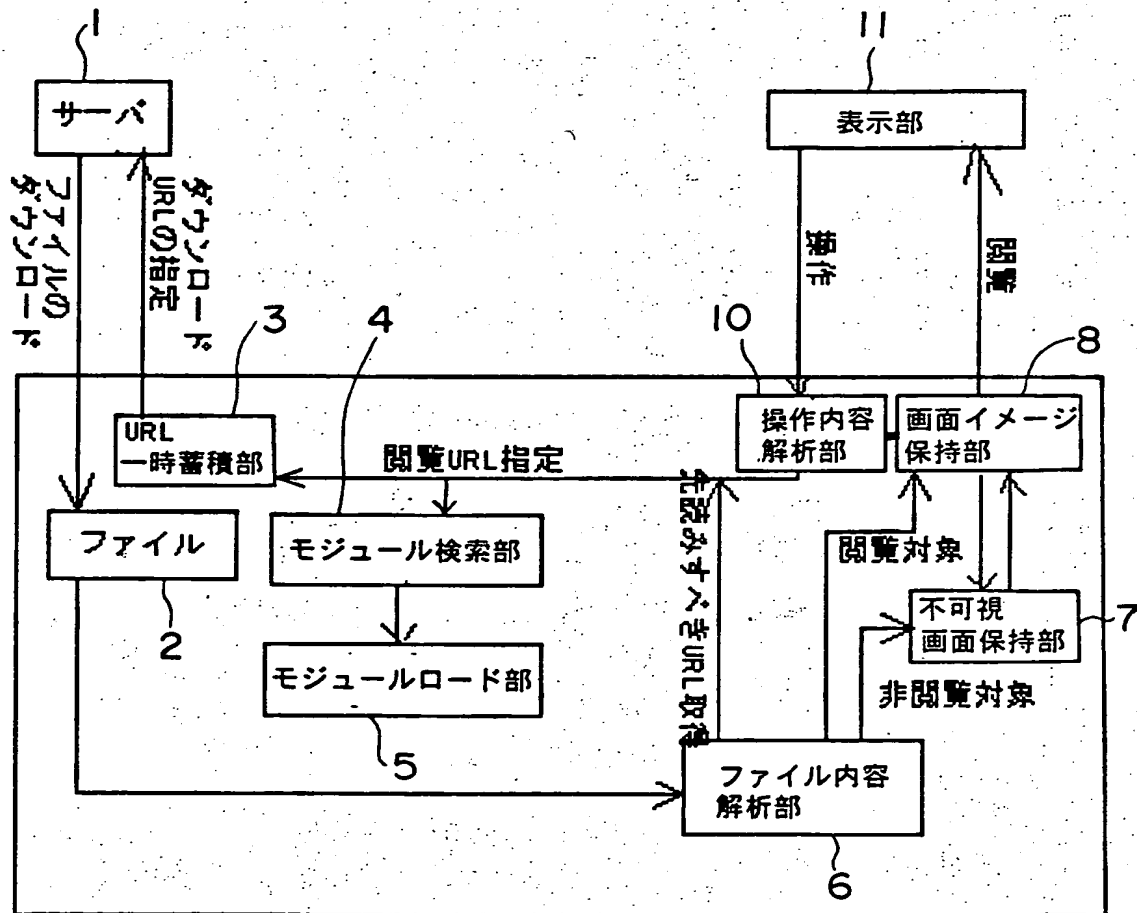
1 7 0 1 表示画面

1 8 0 1, 1 8 0 1 a カーソル

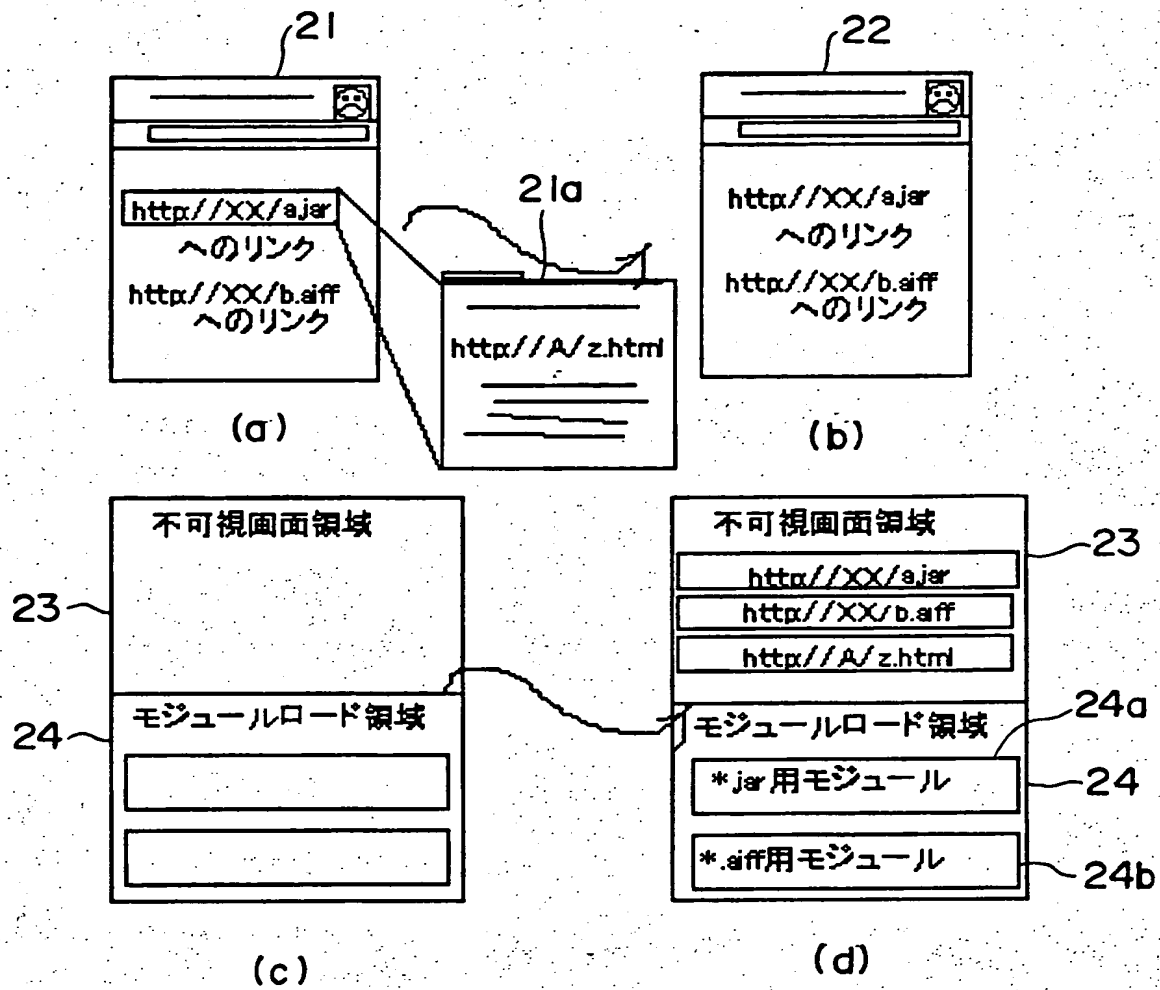
1 8 0 2 先読みすべきURL (別ファイル)

【書類名】 図面

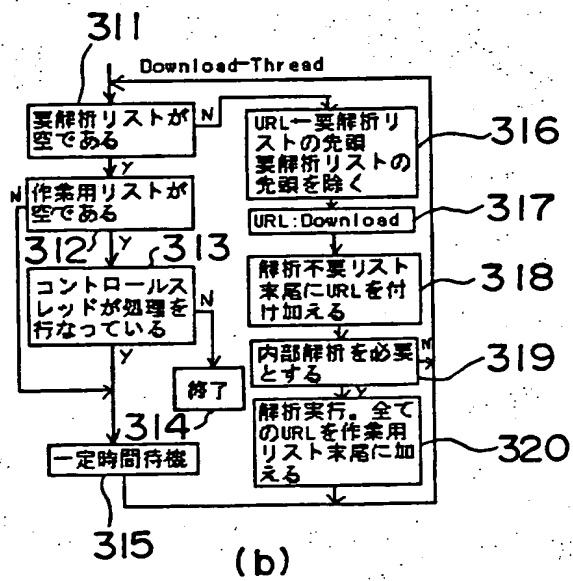
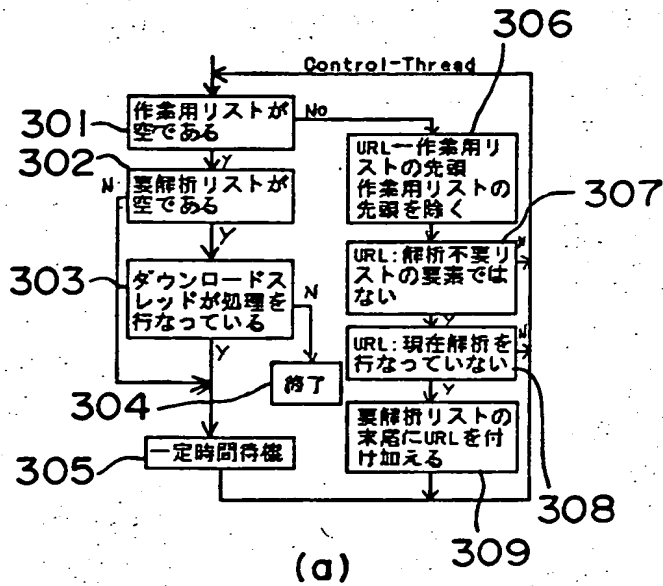
【図 1】



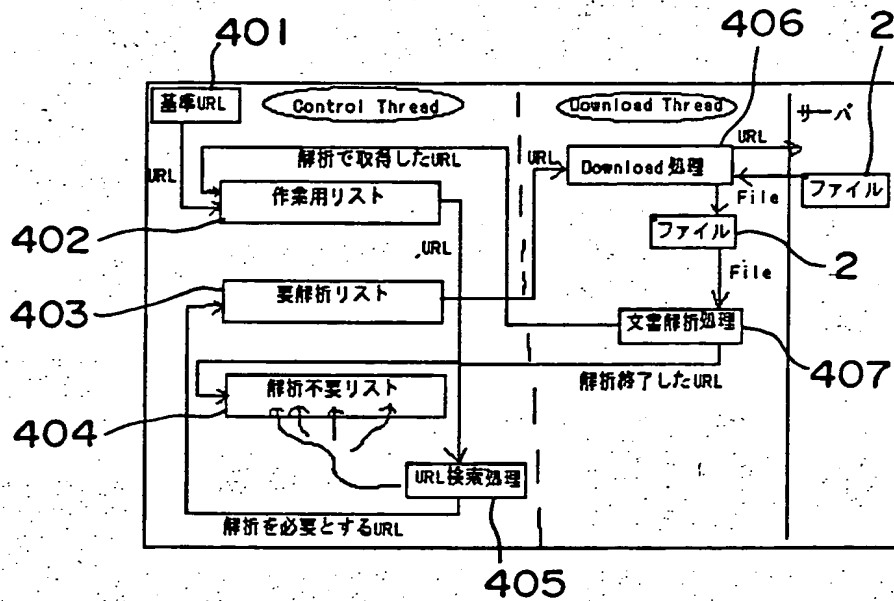
【図 2】



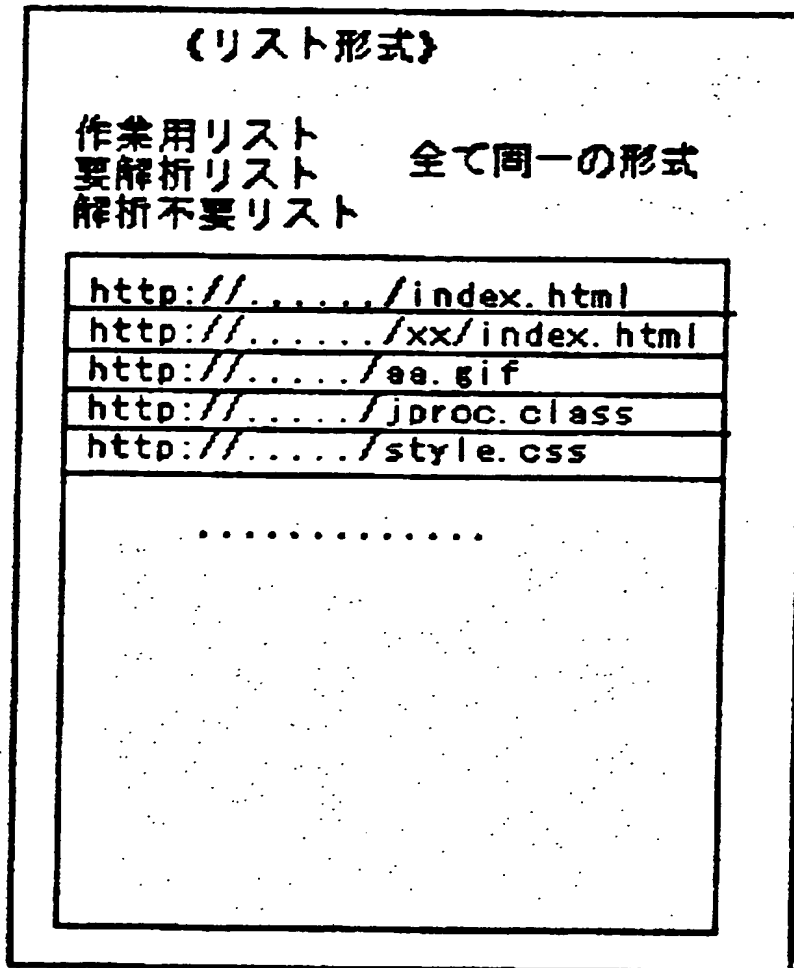
【図3】



【図 4】



【図5】



【図 6】

magic
 version
 constant_pool_count
 constant_pool
 access_flags
 this_class
 super_class
 interfaces_count
 interfaces
 fields_count
 fields
 methods_count
 methods
 attribute_count
 attributes

この部分のみ
解析すればよい

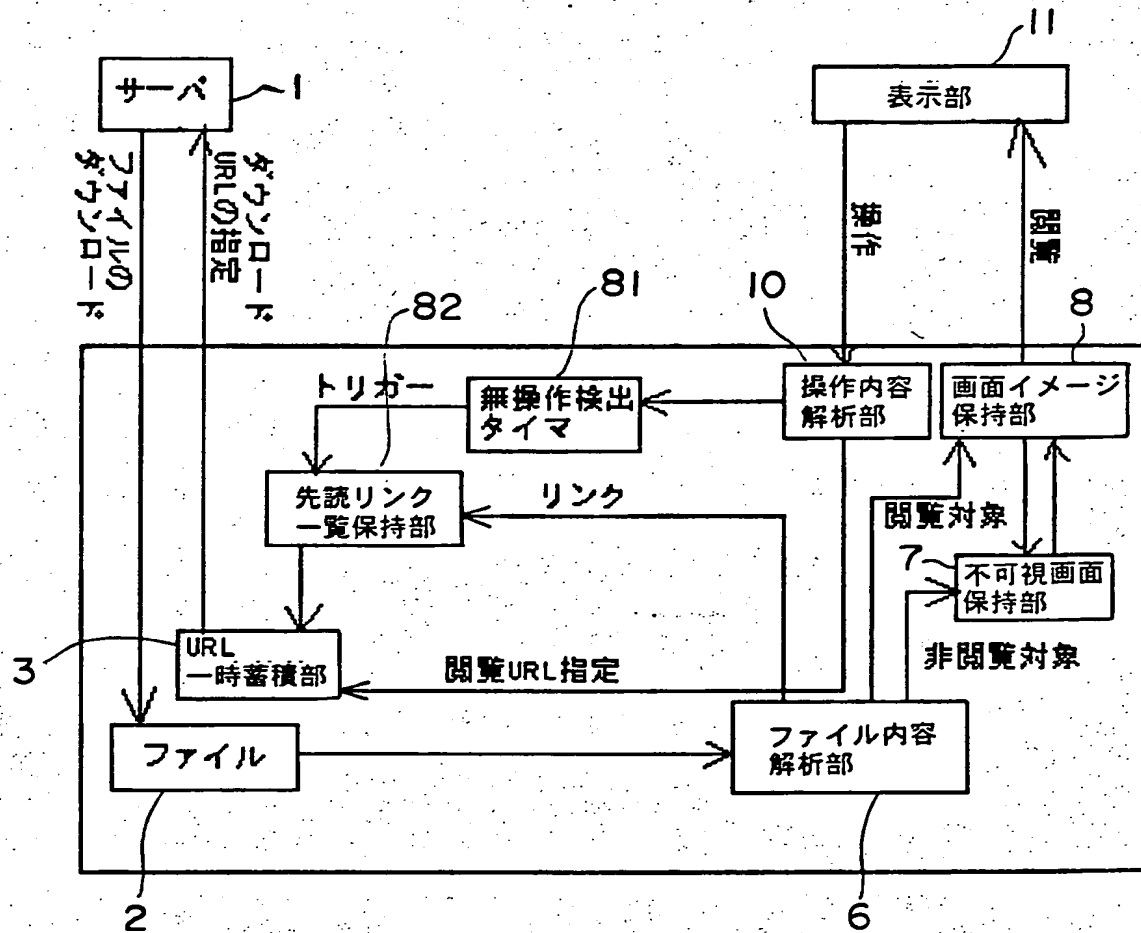
【図 7】

```
<script language="JavaScript">
<!--
function goDROPDOWN(){
if (document.form1.elements[0].selectedIndex == 0) location = "http://www.fujitsu.co.jp/"
else if (document.form1.elements[0].selectedIndex == 1) location =
"http://www.yahoo.co.jp/"
else if (document.form1.elements[0].selectedIndex == 2) location =
"http://www.miti.go.jp/"
}

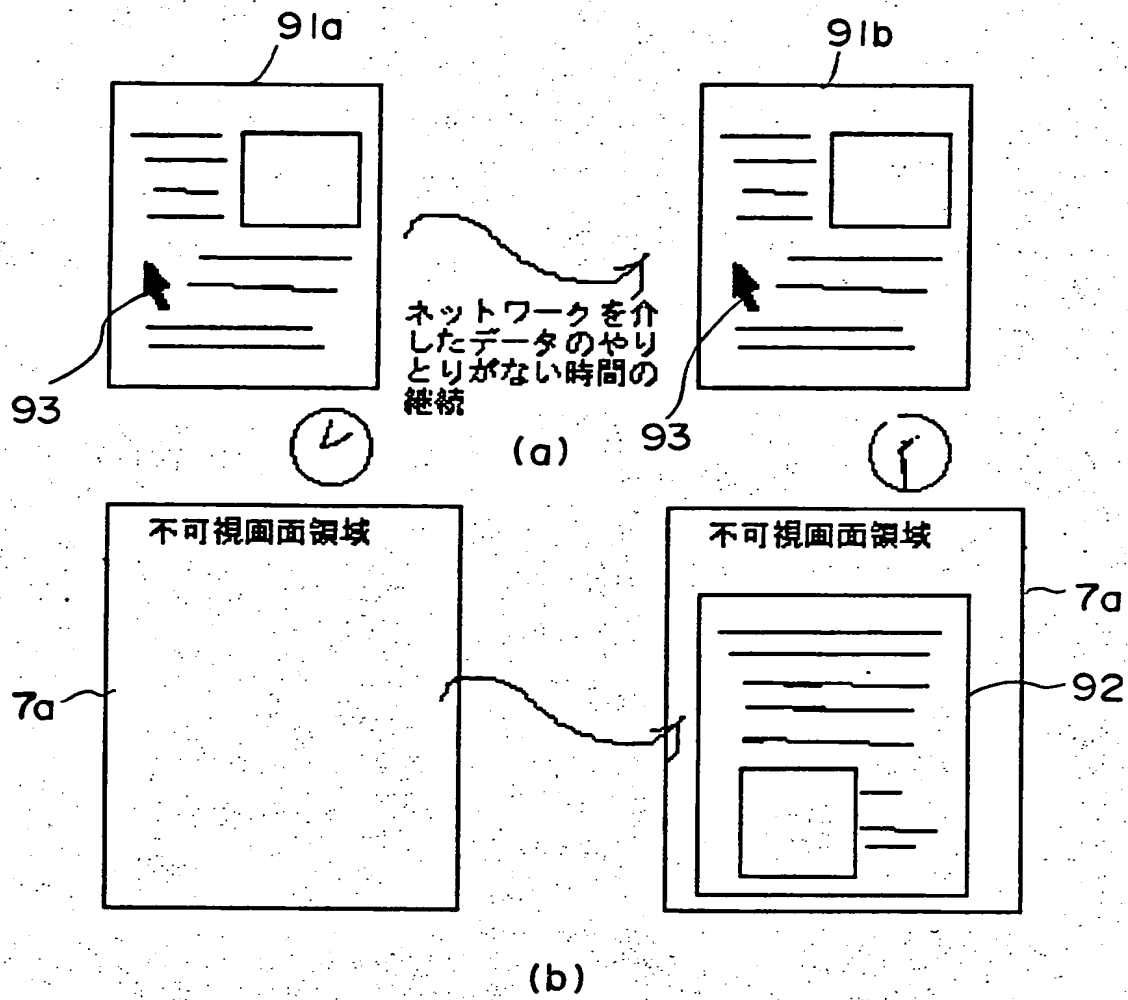
function goLISTBOX(){
if (document.form1.elements[2].selectedIndex == 0) location = "http://www.fujitsu.co.jp/"
else if (document.form1.elements[2].selectedIndex == 1) location =
"http://www.yahoo.co.jp/"
else if (document.form1.elements[2].selectedIndex == 2) location =
"http://www.miti.go.jp/"
}

function home(){
location = "/home.html"
}
// ->
</script>
```

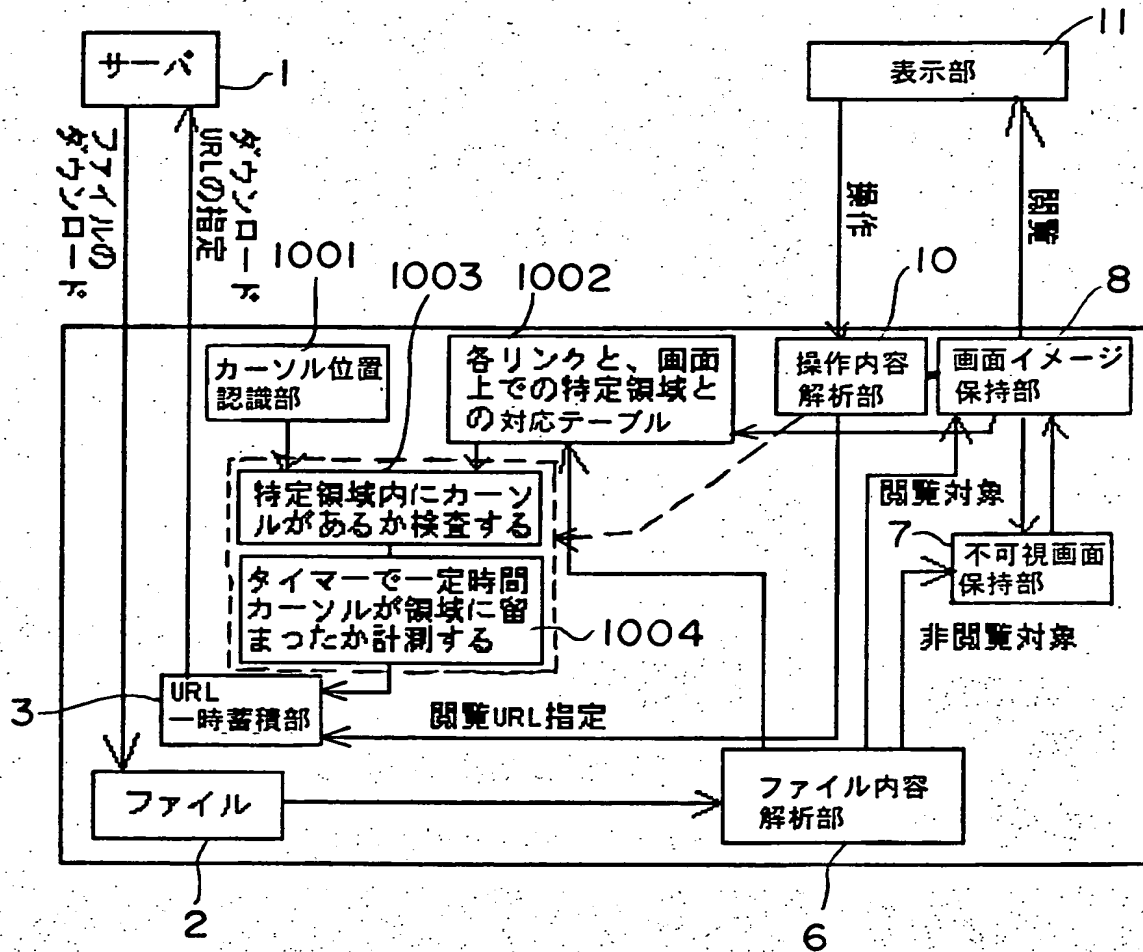
【図 8】



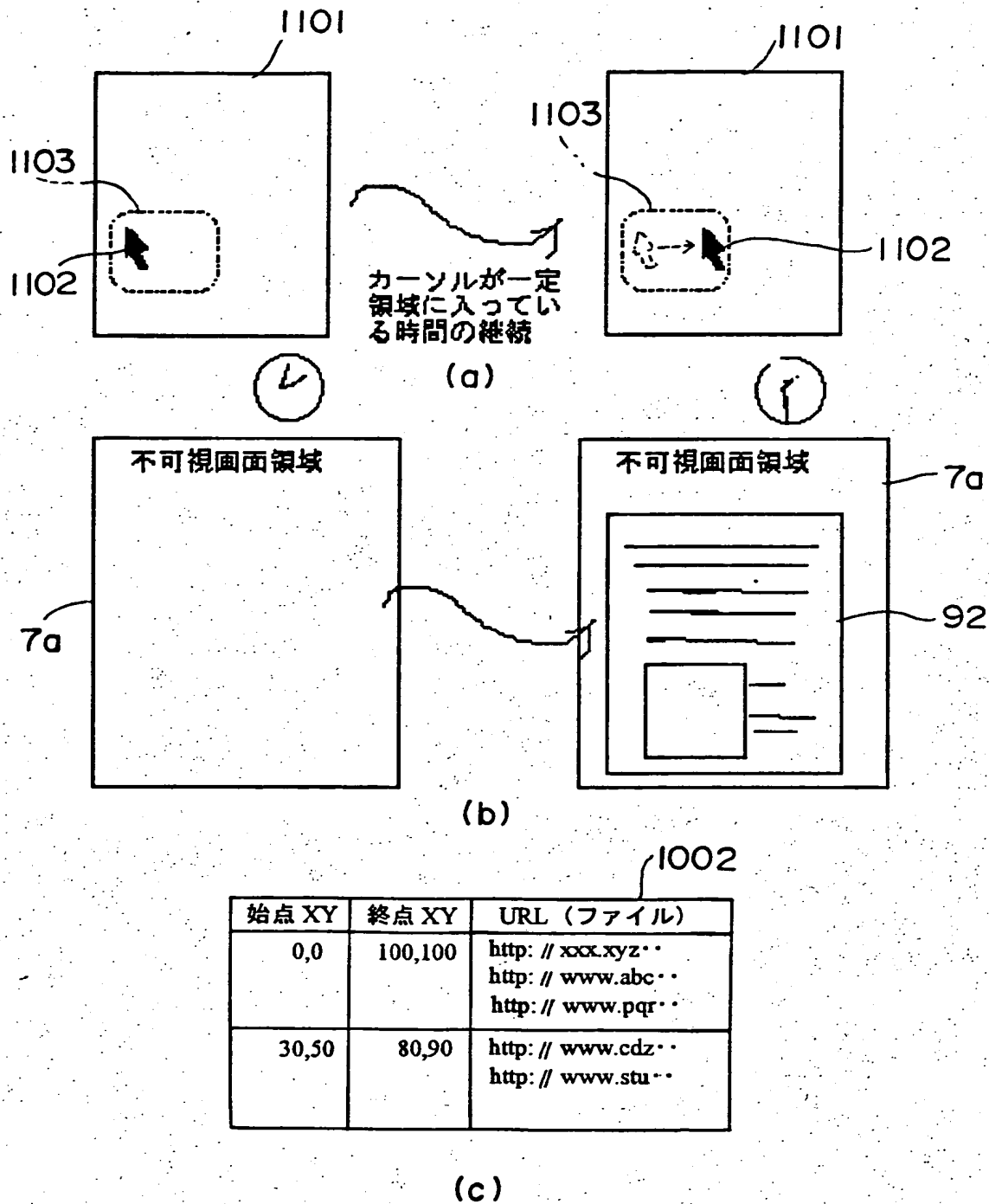
【図 9】



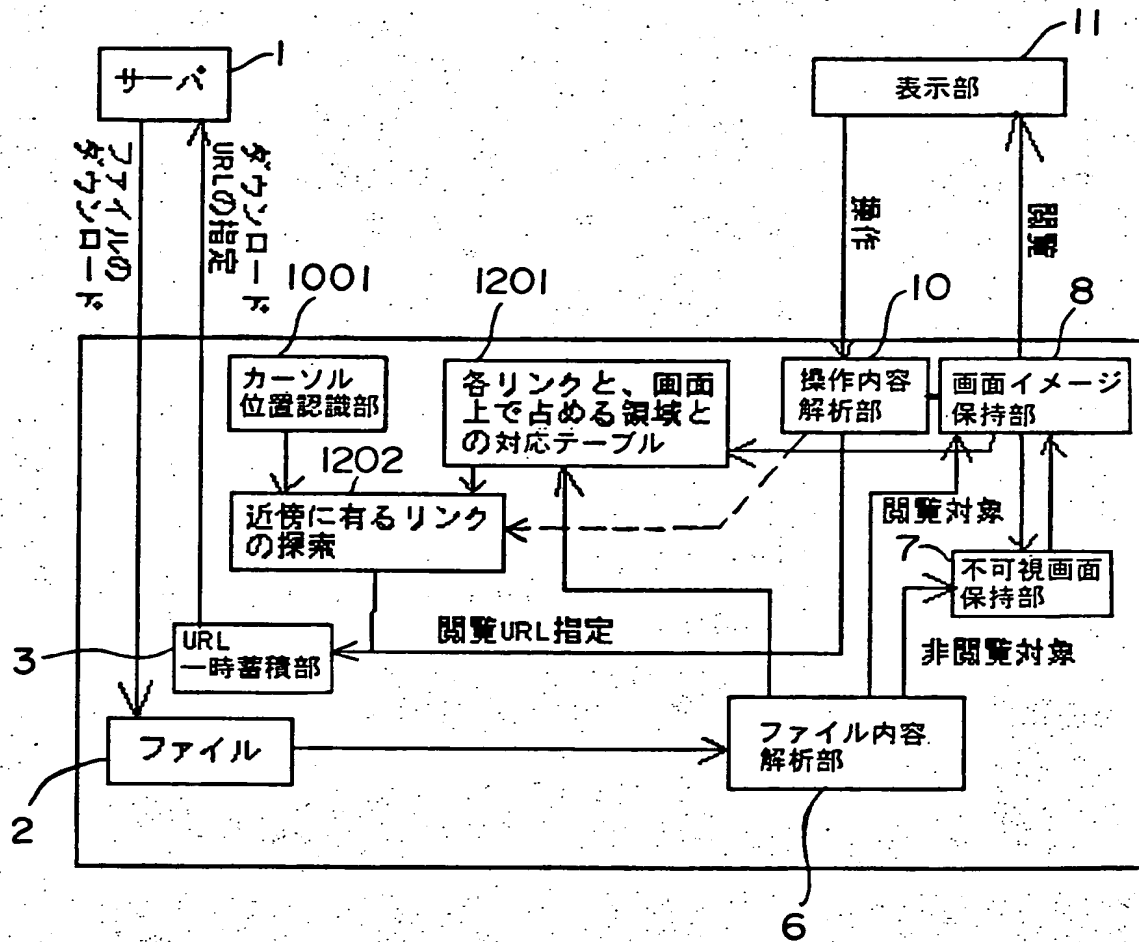
【図10】



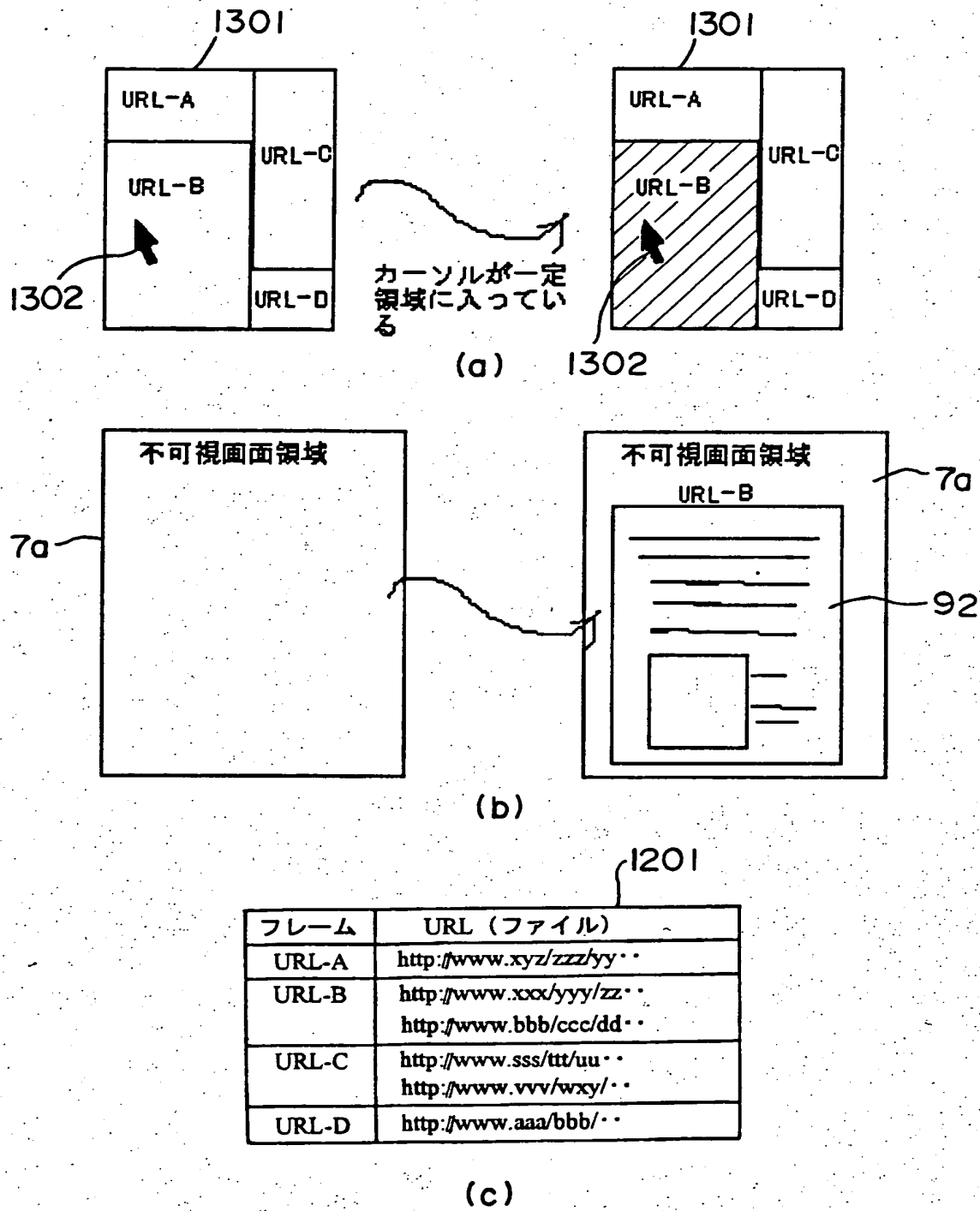
【図 11】



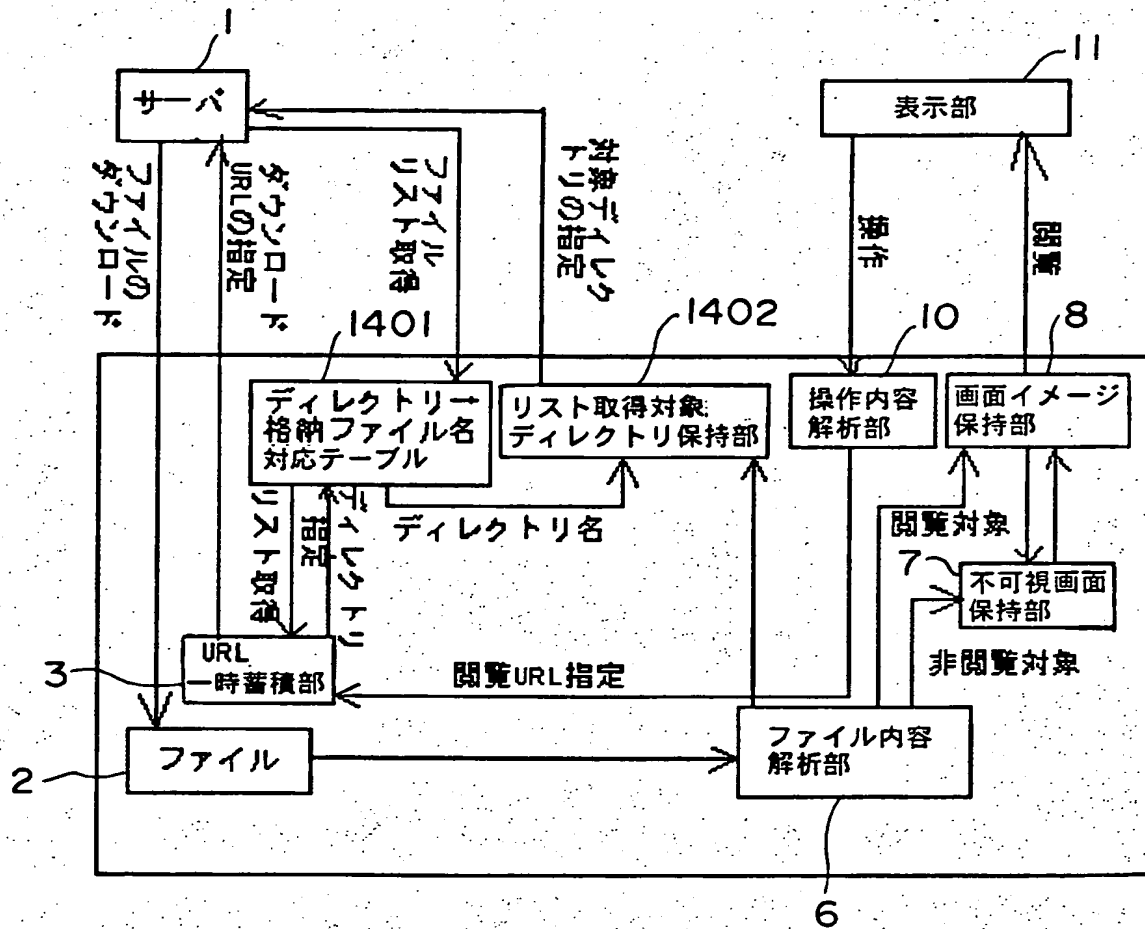
【図12】



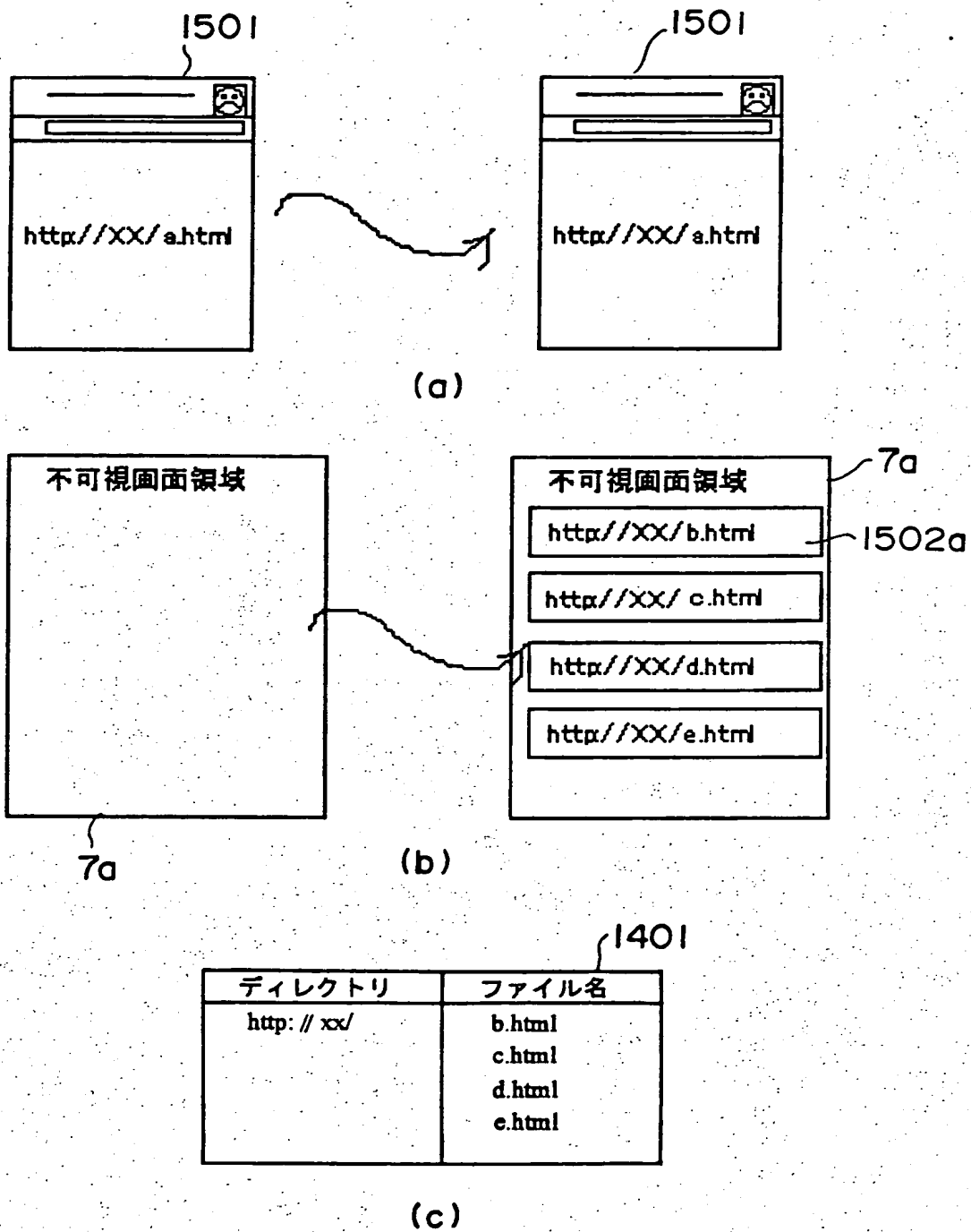
【図 13】



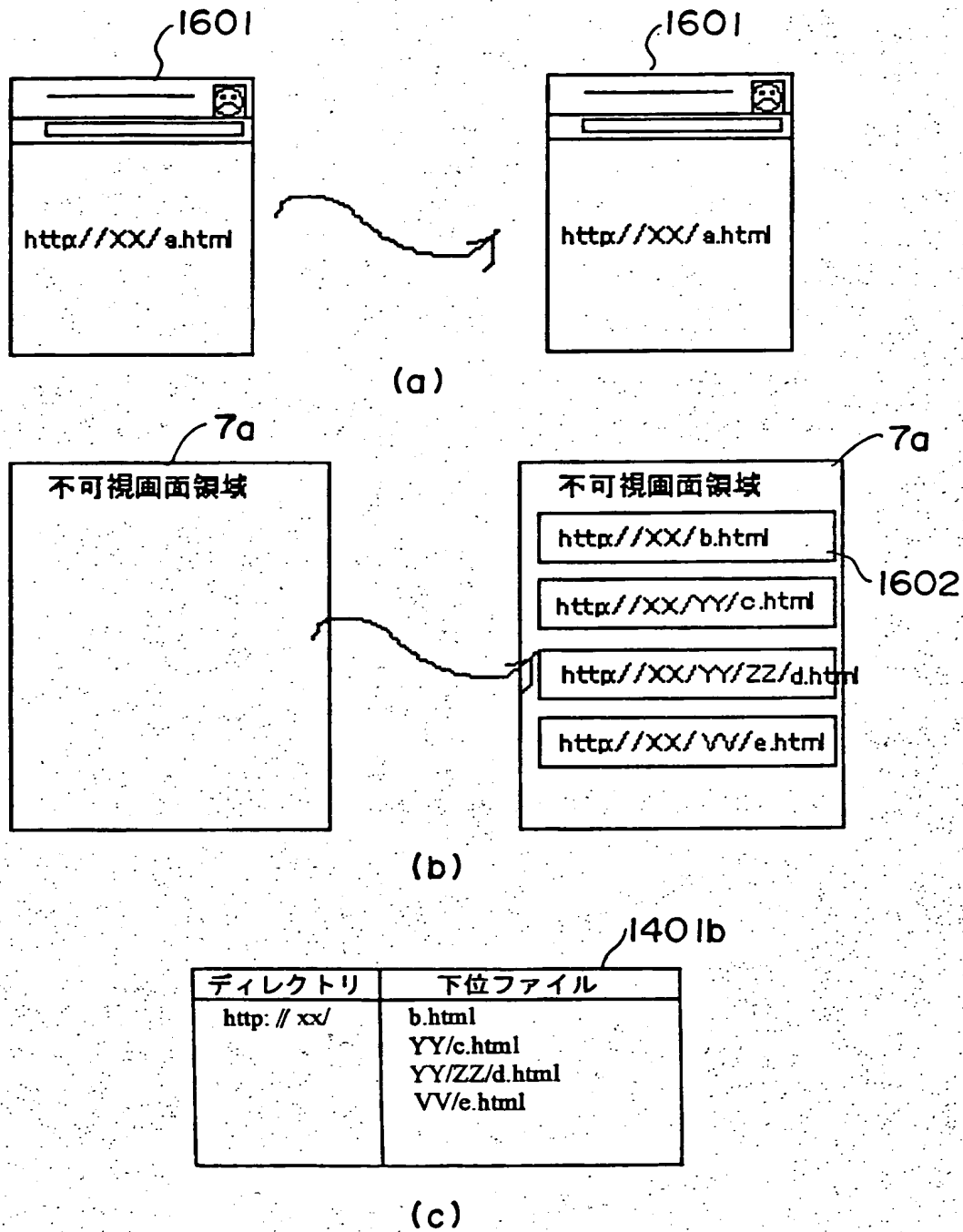
【図14】



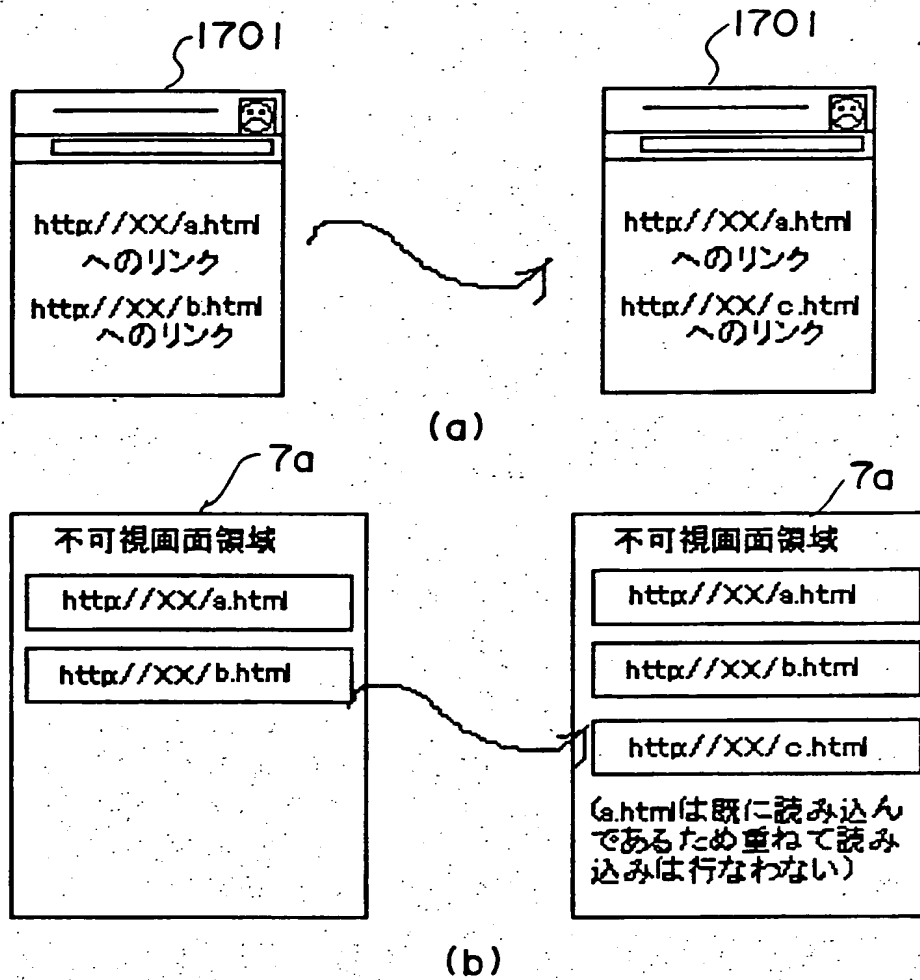
【図 15】



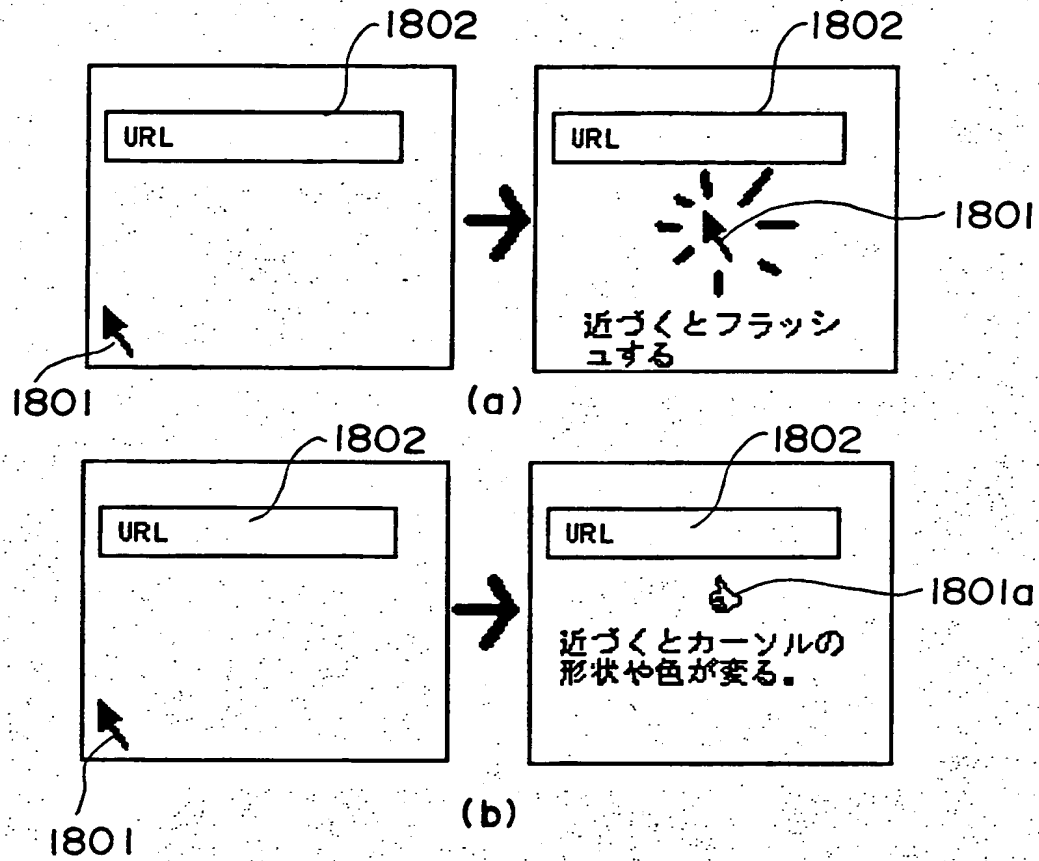
【図 16】



【図 17】



【図 18】



【書類名】 要約書

【要約】

【課題】 インターネットのWWW閲覧を効率的に行う

【解決手段】 インターネットブラウザ等を用いてHTMLファイル等の表示制御ファイルで規定されたリソースファイルを閲覧する際に、このリソースファイルを解析して、リソースファイルに記述された別ファイルをユーザーから指示がある前にサーバからダウンロードし、当該別ファイルを不可視画面領域にロードしておくことにより、ソースページの記述内容を解析して当該ファイルの特性に応じて必要なモジュールを事前ロードしたり、カーソル位置や移送状態等から閲覧者の意志を推測して特定の条件の下にファイルの先読みを実行することにより、閲覧者の意志に沿った先読みを可能にする。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日	1996年 3月26日
[変更理由]	住所変更
住 所	神奈川県川崎市中原区上小田中4丁目1番1号
氏 名	富士通株式会社